For More Information

We suggest several places to find more information about Pascal and the environment in which Oregon Software's products are used. Many of these books are available from Oregon Software. Prices are subject to change without notice.

Oh! Pascal by Doug Cooper and Mike Clancy.
An easy-to-read Pascal course for the novice programmer. (W. W. Norton, $15.95)

Programming in Pascal by Peter Grogono.
A good course in standard Pascal, with lots of sample programs for experimentation. (Oregon Software supplies one copy to each new customer.)

Introduction to Pascal by Rodnay Zaks.
A complete tutorial on Pascal designed to be read and understood by everyone. (SYBEX Inc., $14.95)

A User Guide to the UNIX System by Rebecca Thomas and Jean Yates.
A beginner's tutorial on the UNIX operating system. (OSBORNE/McGraw-Hill, $15.99)

UNIX Programmer's Manual Seventh Edition, Volume 2A.
A collection of tutorials on the use of the operating system and its utility programs; Brian Kernighan's descriptions of the system and the editor are especially helpful for new users. (Bell Laboratories, $60.00 for Volumes 2A and 2B)

Pascal User Manual and Report by Kathleen Jensen and Niklaus Wirth.
The first definition of standard Pascal. (Oregon Software supplies one copy to each new customer.)

Algorithms + Data Structures = Programs by Niklaus Wirth.
A study of programming data structures, beginning with records, arrays, and sets — the fundamental structures — and progressing to those structures that are changed in value and structure by program execution. Full-length sample programs illustrate the stepwise refinements involved in developing Pascal programs. (Prentice-Hall, $25.95)

Structured Programming by Dahl, Dijkstra, Hoare.
Three monographs on methodologies of concept modelling (Dijkstra), data structuring (Hoare), and structured, hierarchical programming (Dahl and Hoare). (Academic Press, $20.00)

Elements of Programming Style by Kernighan and Plauger.
A practical demonstration of the principles of good programming and the use of common sense. The authors criticize and rewrite sample programs from various texts on programming. (McGraw-Hill, $3.95)

Systematic Programming: An Introduction by Niklaus Wirth.
A description of the technique of constructing and formulating algorithms in a systematic manner, intended as general mathematical background rather than practical study of coding. (Prentice-Hall, $23.95)

UNIX Programmer's Manual, Seventh Edition, Volumes 1 and 2B.
Volume 1 is the *Programmer's Reference*, containing explanations of all programs that make up the UNIX operating system. Volume 2B is the *Programmer's Manual*, a collection of papers on various aspects of installing and using the operating system for program development. Topics include various compilers, the assembler, and

the UUCP network. (Bell Laboratories, $100.00 includes Volumes 1, 2A, and 2B)

Concurrent Euclid, the UNIX System, and Tunis by Holt, Graham, Lazowska, Scott.
An introductory text on concurrent programming, the techniques used to design and implement operating systems, computer networks, real-time control and embedded microprocessor systems. (Addison-Wesley, $15.95)

Pascal Newsletter
Published quarterly, Oregon Software's *Pascal Newsletter*, which contains status reports on all of our Pascal products, announcements of new versions of software and new products, and various technical articles.

Temporarily, the newsletter is the sole publication for the Oregon Pascal Users Society (OPUS), an organization dedicated to the sharing of information between Oregon Software and its customers. Oregon Software is not affiliated with OPUS, but we encourage its activities and provide space for an OPUS column in our Pascal Newsletter, until OPUS begins to publish its own newsletter. OPUS membership is free. To join, write:

Oregon Pascal Users Society (OPUS)
Bruce Williams
c/o EOCOM
15771 Redhill Ave.
Tustin, California 92680
(714) 730-5051, ext. 302.

The Pascal Newsletter
Published by the Pascal Users' Group, *The Pascal Newsletter*, is available at $10 for a one-year subscription. Contact:

Pascal Users' Group
2903 Huntington Rd
Cleveland, Ohio 44120.

Index

Style note: Page numbers in boldface (**6**) indicate the defining use of the term.

Documentation Evaluation Report

Pascal–2 V2.1 User Manual Update Package No. 2
PDP-11/RT-11

Good documentation is as important as good software. We at Oregon Software are well aware that you expect both, so we value your responses. Use this form to write down comments and suggestions, which will help us improve the quality and usefulness of our publications. If you require a written response, submit your comments on a Trouble Report.

Rate the following items on a scale of 1 to 5, with 5 being the highest rating.

____ Initial impression
____ Organization
____ Ease of reading
____ Ease of finding information
____ Accuracy
____ Ease of updating the manual

What changes in the documentation, in your opinion, are most useful? ____________________

____
____
____
____
____

What aspects of the update package need improvement? ____________________

____
____
____
____
____

What errors have you found in this update package? Include page numbers. ____________________

____
____
____
____
____
____
____
____

Name ______________________________ Site # ______________

Company ____________________________ Date ______________

Mailing Instructions

Domestic. After filling out the report, cut along the vertical dotted line to remove the punched holes. Then fold this page so these instructions are hidden and the BUSINESS REPLY MAIL permit faces out. Secure the loose fold with staple or tape, then mail.

International. Please enclose this page in an envelope and return it to your distributor. If you do not have a distributor, return the report directly to us. (Unfortunately, we cannot supply prepaid postage for overseas respondents.)

Thank you in advance for your response.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. A407 PORTLAND,OR.

POSTAGE WILL BE PAID BY ADDRESSEE

OREGON SOFTWARE, INC.
Attn: Documentation Comments
6915 SW Macadam
Portland, OR 97219

9EC-92

Release Package Checklist

SOFTWARE: Pascal–2 Development System
VERSION: V2.1D
OPERATING SYSTEM: RT-11
DATE OF RELEASE: January 15, 1985

This release package contains the following marked items. If a discrepancy exists between the marked items and the contents of your release package, please contact Oregon Software at (503) 245-2202 immediately.

☒ Pascal–2 software on:
- □ Magnetic tape. □ 800 bpi □ 1600 bpi
- ☒ Floppy disk.
- □ Cartridge disk.

☒ *Pascal–2 User Manual*, Second Edition.
☒ Documentation Update Package, Update No. 2.
☒ *Release Notes*, including the Installation Guide.
☒ *Standard Pascal User Reference Manual* by Doug Cooper.
☒ *Programming in Pascal* by Peter Grogono.
☒ An assortment of *Pascal Newsletters*.
☒ Extra Trouble Report forms.
□ Extra "Request to Amend" forms.
□ License agreement.

Release Notes

Pascal–2 V2.1D for RT-11

January 15, 1985

The software described by this publication is subject to change without notice. Oregon Software assumes no responsibility for the use or reliability of any of its software that is modified without the prior written consent of Oregon Software.

Oregon Software holds right, title, and interest in the software described herein. The software, or any copies thereof, may not be made available to or distributed to any person or installation without the written approval of Oregon Software.

This publication, or parts of it, may be copied for use with the licensed software described herein, provided that all copies include this notice and all copyright notices.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in sub-division (b)(3)(ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013 of the Federal Acquisitions Regulations (FARs).

Name of Contractor and Address:

Oregon Software, Inc.
6915 S.W. Macadam Avenue
Portland, Oregon 97219
Phone: 503-245-2202

Pascal–2 V2.1D/RT-11 Release Notes

The information contained in this document describes the Pascal–2 V2.1D release package. In these notes you will find:

- A list of problems ("bugs") we've fixed in this release.
- Notes on changes in the documentation and on how to update the user manual.
- Miscellaneous notes of interest to Pascal–2 users.
- The Installation Guide.

These release notes should be read before attempting to install the compiler and utilities on your system. The "Documentation Notes" section of the release notes should be read before attempting to insert the change pages from the update package into the user manual.

Style Notes

This document follows these style conventions:

Text:

Pascal reserved words, predefined symbols, switches and compiler directives are in boldface typewriter: **begin**, **write**, **%include**, **nomain**. Portions of examples referred to in the text appear in boldface typewriter (the type style used in examples). Program, system and file names are in upper-case letters in the same type style as the text: SAMPL, VMS, SAMPL.PAS.

Program Examples:

Commands that you should enter are in underlined boldface typewriter: **RUN EX**. These commands assume a carriage return at the end.

Program Listings:

The Pascal–2 compiler accepts any combination of upper-case and lower-case characters. Examples in this manual have Pascal words in lower case and have user-defined words with an initial capital letter and other capitalization as needed for readability, as shown in this program fragment:

```
procedure Show;
begin
  SomeUserAction;
  writeln(Result);
end;
```

Terminology:

We use standard terms as they are used in documents describing the RT-11 operating system.

Changes in the Software

Version 2.1D is primarily a maintenance up-date; changes in software between Versions 2.1D and 2.1C consist for the most part of fixes for previously known problems.

Problems We've Fixed

Version 2.1D corrects the following problems:

Record field as a parameter

The compiler generated bad code when certain nested record fields were passed as a parameter.

Negative stack offset for common subexpressions

Depending upon the location of a variable's declaration, two identical calculations involving certain functions, including **sin** and **cos**, could have different results. The library routines for the functions caused the compiler to reference a negative offset from the stack pointer.

Wrong results from integer comparison

A comparison between an integer and a subrange that is an element of a packed record produced an incorrect result.

Incorrectly addressed real variable

Certain globally allocated real variables were incorrectly addressed. As a work-around, reducing the size of the program seems to remove the error.

Dynamic string package errors

The **Insert** routine printed the error message **Array subscript out of bounds** when the combined length of the two input strings exceeded the target string's length. Now, the insert string is concatentated onto the end of the target string and truncated.

The **Concatenate** routine reported errors when the string to be concatenated overflowed the target string. The input string is now truncated to fit.

%Page directive didn't work

On listings, **%page** incremented the page number incorrectly.

Reserved instruction trap error

Using a **packed** record containing integers in a nested procedure call caused a reserved instruction trap.

Illegal instruction gave "compiler writer error"

Using certain illegal instructions resulted in the error message for internal problems instead of the appropriate error message.

No error message

The compiler failed to give an error message for attempts to pass an element of a packed structure as a variable-parameter.

The compiler incorrectly used R0 for procedure calls

The compiler failed to save the contents of the R0 register after **nonpascal** procedure calls.

Memory protection violation

A memory protection violation occurred in isolated instances when the value of R0 was not initialized before adding the offset.

Sets generated bad code

Set constructor expressions of the form `[var1..var2]` generated bad code at times.

Set type produces wrong results

The declaration and statement

```
type  a = set of (one,two,three);
var   b:a;

begin
  b:= [one,three];
  write(b * [] <= []);
end.
```

produced a set of 256 items merged with a set of 3 results.

Function returns incorrect result

The compiler incorrectly changed bit lenghts to byte lengths when a function returned a structured type.

Bit optimisations incorrect

Compiler optimization of certain user-defined procedure calls produced an incorrect sequence of execution.

PB formatter rejects "nonpascal"

The formatter failed to recognize the **nonpascal** directive. It now treats **nonpascal** exactly like the **external** directive.

String comparison range wrong

For string comparisons, `ord(char)` was in the range `0..255` instead of `-128..127`. The range is now correct.

Unsigned characters treated as signed

The compiler treated 8-bit characters as if they were signed numbers when the character was optimized.

Assignment statements caused failure

A complicated series of assignment statements involving arrays of type **real** previously failed.

Compiler consistency checks reported

Certain instances of `undeleted temps in procedure...` were fixed in 2.1D.

Linkage failure during installation

A missing continuation character in the file UTILS.XM, which builds the utilities, caused a failure at the point where XREF is linked.

Misleading error message for nonexistent files

The compiler printed the error message "Unknown Pascal run-time error" for an attempt to compile a nonexistent file or a file containing a `$include` of a nonexistent file.

Debugger overlays do not work for XM monitor systems

XMDBG.COM, the XM monitor command file that links Debugger modules with a Pascal program, now correctly overlays Debugger modules.

Documentation Notes

The documentation for Pascal–2 Version 2.1D includes the second edition of the *Pascal–2 User Manual*, which documents the enhanced V2.1 software, and Update Package No. 2, documenting the 2.1D features. (Update Package No. 1 was previously distributed and all changes then noted have been incorporated into the manual.) We have expanded a number of sections, including "External Modules," "Resident and Cluster Libraries," and "The Debugger Guide." The expanded sections provide new information and clarify and correct earlier material, at user suggestion. Update Package No. 2, records these changes and is made up of "change pages" for insertion into the user manual at the appropriate place. The update package's cover sheet should be inserted into the manual just before the first contents page, to keep a history of changes to the manual.

The *Release Notes*, containing the Installation Guide, is a companion document of the *Pascal–2 User Manual* and should be kept with that manual for reference.

Miscellaneous Notes

Official ISO Standard for Pascal

The Pascal language now officially has an international standard. The International Standards Organization, in a vote tallied this summer, adopted a standard language definition for Pascal. The action followed earlier adoption of an American standard that is a subset of the international one, lacking only conformant array parameters.

The sequence leading to adoption of the Pascal standard began in December 1982, when ANSI and IEEE agreed on the American standard, identical to the international draft standard except for conformant array parameters. At the same time, the Joint Pascal Committee of ANSI and IEEE recommended adoption of the international standard Level 1 (including conformant array parameters) to the U.S. committee known as X3J9, which then voted "yes" on the international standard at the next meeting. Previously, the U.S. was one of three "no" votes. This time, the ISO standard passed with no dissenting votes and one abstention.

Peppered throughout the Language Specification of the user manual are references to the "draft" standard. The word "draft" can be ignored. (Change pages were not issued for this change.)

Packing Problems

Several users have uncovered bugs in Pascal–2's packing of records and arrays. We believe that it may be necessary to modify the packing algorithm slightly in version 2.1E in order to correct these problems. If such a change is necessary, users who have programs that involve the reading and writing of packed records or arrays may have to convert their data files to the new format in order to use V2.1E. This new version will be released in roughly six months; if the packing conversion is necessary we will make every effort to clearly document when and how to follow the conversion process.

Stack Overflows

Especially large stack overflows, those that extend beyond the heap and into the program code sections of memory, have the potential for causing serious problems. In some cases it is possible for the error to prevent the appropriate error message from being printed or the program from properly terminating. In some rare instances, the condition can even lead to the disruption of the computer's operating system. It is the programmer's responsibility to avoid such excessively large overflows by controlling the size of local variables and value parameters passed to procedures.

EXITST Creates a Walkback for 'Severe Error' Status (4)

The Exitst procedure is a support library routine that sets the termination status of a program and stops the program when a "severe error" status is detected. The procedure's integer argument determines the termination status for any program that calls it. When a "severe error" status of 4 is passed, the procedure also invokes the post mortem analyzer to create a walkback of the program execution from the point of failure.

Pascal–2 V2.1D/RT-11 Installation Guide

This guide describes the way to load the RT-11 Pascal–2 software on your system. The V2.1 software runs on RT-11 V4 and V5.

To install Pascal–2 V2.1 for RT-11, follow the steps below:

1. Copy all of the Pascal–2 files to the system device (SY:);
2. Select a compiler depending on your system monitor;
3. Select a run-time library depending on your processor hardware options;
4. Compile the Pascal–2 utility programs;
5. Set the compiler's listing file page size to a value other than 66 octal (optional);
6. Extract the Debugger modules from the support library for use in overlaying the Debugger against user programs.
7. Delete files no longer needed (optional). See Appendix B for a sample deletion command.

These steps are described in detail in the following paragraphs, and are illustrated by an example that can be found in Appendix B of this guide. The installer should read this guide in its entirety before attempting to install the software. Upon successful completion of the installation, the Pascal–2 system is fully operational as described in the *Pascal–2 User Manual.*

This guide contains two appendices. Appendix A lists the files contained in the release media. Appendix B shows a typical XM installation.

Copying the Pascal–2 Files to the System Device

Copy all of the distribution files to the system device (SY:) using the ASSIGN and COPY commands, as shown below:

```
.ASSIGN SY DK  ——————————————————— assigns SY: as default device
.COPY MT0: SY:  ——————————————— copies magtape to system device
```

If your system disk is a floppy disk, or is otherwise limited in available space, you should first read the following sections and select the files that are necessary for your system. Then copy those files to your system. The minimal system requires one compiler file and one library file. You may wish to build more than one system disk and, for example, install the compiler, library, and Debugger on one disk and the utility programs on a second system disk.

Selecting a Compiler for Your System Monitor

There are two compilers supplied with Pascal–2. Your choice of a compiler depends on the version of the RT-11 monitor you intend to use. There are four possibilities: the Base-Line (BL) monitor, the Single-Job (SJ) monitor, the Foreground-Background (FB) monitor, and the eXtended-Memory (XM) monitor.

If you are using either the BL or the SJ monitor, choose the compiler called SJ.SAV. If you use the XM monitor, select the compiler called XM.SAV.

The FB monitor does not leave sufficient memory to run the Pascal–2 compiler. If you are using the FB monitor, you must switch to either the SJ or XM monitor (using the BOOT command) before compiling a Pascal program. Once compiled, programs may be linked and run under the FB monitor.

The XM compiler should be chosen over the SJ compiler where possible because it uses the extended memory or "virtual" overlay capability and gives faster compilations.

When you have selected a compiler file, copy it to SY:PASCAL.SAV with the following command, which assumes the default device is SY:.

```
.COPY SJ.SAV PASCAL.SAV  ———————————————————— installs SJ compiler
        or:
.COPY XM.SAV PASCAL.SAV  ——————————————————— installs XM compiler
```

You may then delete SJ.SAV and XM.SAV, or you may leave them on your system disk for use under their respective monitors. (See Appendix B.)

Selecting a Run-Time Library

There are four run-time libraries supplied with Pascal-2, one for each combination of processor instruction sets. Choose the library that matches the configuration of the processor that will run your compiled programs.

The possible configurations are:

- FPP — a processor with the Floating Point Processor instruction set. The FPP is standard equipment on the PDP-11/60 and optional on all new PDP-11's and the LSI-11/23. If your processor includes the FPP, select the LIBFPP.OBJ library.
- FIS — the Floating Instruction Set. The FIS hardware is an option available for the LSI-11, LSI-11/2 and some older PDP-11 processors. If your processor has FIS, select the LIBFIS.OBJ library.
- EIS — Extended Instruction Set, for hardware support of multiply, divide, and long shift instructions. EIS is standard equipment on all new PDP-11 and LSI-11/23 processors and an option available for all older LSI-11's and PDP-11's. If your processor has neither FPP nor FIS, but does have EIS, then select the LIBEIS.OBJ library.
- For processors with no extended or floating instructions, select the LIBSIM.OBJ library. This library operates on any LSI-11 or PDP-11 regardless of its actual configuration, but does not take advantage of any optional hardware.

After selecting a library file, copy it to SY:PASCAL.OBJ with the following command or one similar to it, depending on the processor configuration. In the command, SY: is the default device.

```
.COPY LIBFPP.OBJ PASCAL.OBJ  ————————————————————— FPP library
```

You may then remove the other library files, or leave them on the system for use with other configurations. (See Appendix B.)

Compiling the Utility Programs

Five Pascal-2 utility programs are supplied in source form and are automatically compiled and linked by one of two command files, UTILS.SJ and UTILS.XM. UTILS.SJ prepares the utility programs for execution under the SJ, FB and BL monitors; UTILS.XM prepares them for execution under XM. Feel free to examine these command files before executing it indirectly using one of two commands:

```
.@UTILS.SJ
    or:
.@UTILS.XM
```

Upon completion of UTILS, the utility programs are available for use as described in the Utilities Guide of the *Pascal–2 User Manual.*

In UTILS.XM all of the utilities must be linked as "virtual jobs" made up of virtual overlays because their load images are larger than 16K. (The XM monitor can not load a program with a root segment larger than 16K.) The Pascal–2 file START.OBJ is used in the link process to create a null root segment, tricking the Linker into loading the oversized program. The *Pascal–2 User Manual* contains a detailed description of this technique for user programs in Pascal.

When UTILS completes, you may then delete the extra .OBJ and .MAP files left on the disk by the UTILS command file. (See Appendix B.)

Setting the Page Size of the Compiler's Listing File

The command files RTPAGE.COM and XMPAGE.COM, supplied with the Pascal–2 release, provide a means for changing the number of lines per page of the SJ and XM compilers' listing file, respectively. This feature allows for the use of odd-sized printer paper or for special printing needs.

Before executing either command file, you must modify the lines-per-page value in the fifth line of data in the command file. (The first line of data is the file to patch.) The normal page size (in octal) is 66, not counting header lines.

```
! Command file to patch the SJ version of Pascal to change
! the lines per page of a listing file
!
! Change the number nn (octal) to another value, in octal
! that represents the number of lines per page of a listing file
! that you require.
!
R SIPP
SY:PASCAL

3075
2
nn   ------------------------------------------------------------ lines per page
^Y
^C
```

where *nn* is the desired page size in octal. The value '3075' is the base address of the symbol **PAGELN**, and the actual value in the supplied command filemay be different. The value '2' is the byte offset from the base address of the location to patch. Values for the page size and byte offset must not be changed.

After editing the appropriate file, execute it indirectly with the command:

```
@RTPAGE  --------------------------------------------------- for SJ installation
  or:
@XMPAGE  --------------------------------------------------- for XM installation
```

Extracting Support Library Modules for Debugger Overlays

Before users can overlay the Debugger against their programs, the command file EXTRAC.COM must be executed. This command file, supplied with the release kit, extracts from the Pascal support library the Debugger and other key modules used by the XMDBG.COM (for XM systems) and SJDBG.COM (for SJ systems) command files, also in the release kit. (Users can then execute either XMDBG.COM or SJDBG.COM to overlay the Debugger and their Pascal programs.) EXTRAC.COM places the modules onto the system device (SY:), with the extension .DBG. It need only be executed once.

The command file can be executed with the indirect (at-sign) processor, as shown:

```
        .@EXTRAC
```

Installing Pascal-2 With Pascal-1

The Pascal-1 system compiler can be renamed to allow simultaneous use of Pascal-1 and Pascal-2. Then, the files for both systems can be present on the system device without conflict, provided that the compiler/library versions and upgrade levels are themselves compatible. (For example, Pascal-1 V1.2K is compatible with Pascal-2 V2.0K, and V1.3C is compatible with V2.1C; V1.2K, however, is not compatible with V1.3C.

The object library PASCAL.OBJ can be shared by the two systems (as long as the two system versions are compatible); the libraries supplied with Pascal-2 include all of the Pascal-1 routines.

Appendix A: Distribution Files

Compilers

SJ .SAV	Pascal–2 Compiler for SJ monitor
XM .SAV	Pascal–2 Compiler for XM monitor

Object Libraries

LIBFPP.OBJ	Library for processors with FPP and EIS
LIBFIS.OBJ	Library for processors with FIS and EIS
LIBEIS.OBJ	Library for processors with EIS only
LIBSIM.OBJ	Library for base-level processors
VIRJOB.OBJ	Header module for XM virtual jobs
START .OBJ	Null root module for XM virtual overlays

Command Files

EXTRAC.COM	Support library module extractor for Debugger overlays
SJDBG .COM	Debugger overlayer for SJ monitor
XMDBG .COM	Debugger overlayer for XM monitor
RTPAGE.COM	SJ listing-file page size patcher
XMPAGE.COM	XM listing-file page size patcher
UTILS .SJ	SJ, FB and BL utility installer
UTILS .XM	XM utility installer

Utility Programs

PASMAT.PAS	Program formatter
PB .PAS	Pascal Beautifier
XREF .PAS	Cross-referencer
PROCRE.PAS	Procedure cross-referencer
PROSE .PAS	Text formatter
STRING.PAS	String package
PASMAC.MAC	MACRO-11 interface package
SAYERR.PAS	System error message printer
UERROR.PAS	User error reporting module
OPERRO.PAS	Support library error routine
LIBDEF.PAS	Support library definitions
CSITYP.PAS	Command String Interpreter (CSI) type definitions
CSIPRO.PAS	CSI procedure definitons
FIXARG.PAS	CSI argument parser
FIXINC.PAS	CSI %include file handler
FIXOUT.PAS	CSI temporary output manager
SYMDCL.PAS	CSI symbol table declarations
SYMCOD.PAS	CSI symbol table manager
CVTNUM.PAS	CSI command-line number converter

Demonstration Programs

* Vary from release to release

Appendix B: Typical XM Installation

The following steps illustrate the installation of Pascal-2 from magtape on a processor that includes the FPP floating point processor. The Pascal-2 compiler for the extended memory (XM) monitor is selected.

```
.ASSIGN SY DK  ---------------------------------  assigns SY: as the default device
.COPY MT0: SY:
.COPY LIBFPP.OBJ PASCAL.OBJ  ---------------------------  selects the FPP library
.COPY XM.SAV PASCAL.SAV  ------------------------------  selects the XM compiler
.@UTILS.XM  ---------------------------------------------  builds the utilities
R PASCAL  ------------------------------------------------  builds PASMAT
PASMAT/TIME/WORKSPACE=650
! The following link will round up the root to 4K and put the extra
! memory on the heap
R LINK
PASMAT,PASMAT=START,VIRJOB/U:20000//
PASMAT,PASCAL/V:1
//
P$GROW
^C
R PASCAL  ----------------------------------------------------  builds PB
PB/TIME
R LINK
PB,PB=STRAT,VIRJOB/U:20000//
PB,PASCAL
//P$GROW
^C
R PASCAL  -------------------------------------------------  builds XREF
XREF/TIME
R LINK
XREF=START,VIRJOB/U:20000//
XREF,PASCAL/V:1
//
P$GROW
^C
R PASCAL  ----------------------------------------------  builds PROCREF
PROCRE/TIME
R LINK
PROCRE,PROCRE=START,VIRJOB/U:20000//
PROCREF,PASCAL/V:1
//
P$GROW
^C
R PASCAL  -------------------------------------------------  builds PROSE
PROSE/TIME/WORKSPACE=700
R LINK
PROSE,PROSE=START,VIRJOB/U:20000//
PROSE,PASCAL/V:1
//
P$GROW
^C
```

```
.@XMPAGE     ——————————— sets listing page size to value other than 66 octal (optional)

.@EXTRAC     ——————————————————— extracts library modules for Debugger overlaying

.R PIP    ——————————————————————————————— cleans up the system disk (optional)
*SJ.SAV,XM.SAV/D      ———————————————— Remove the compilers, leaving PASCAL.SAV
*LIBFPP.OBJ,LIBFIS.OBJ,LIBEIS.OBJ,LIBSIM.OBJ/D
*PASMAT.OBJ,PB.OBJ,XREF.OBJ,PROCRE.OBJ,PROSE.OBJ/D
*PASMAT.MAP,PB.MAP,XREF.MAP,PROCRE.MAP,PROSE.MAP/D
*^C
```